

Optimization analysis based on a breadth-first exploration for a structural approach of sketches interpretation

Achraf Ghorbel*, Eric Anquetil*, Aurélie Lemaitre†

*INSA de Rennes, †Université de Rennes 2

UMR IRISA, Campus de Beaulieu, F-35042 Rennes

Université Européenne de Bretagne, France

{achraf.ghorbel, eric.anquetil, aurelie.lemaitre}@irisa.fr

Abstract—In this paper, we present an optimized approach, based on a competitive breadth-first exploration of the analysis tree, for an interactive interpretation of off-line sketch. The competitive breadth-first exploration of the analysis tree, allows to compare several hypotheses of interpretation to deal with confusion. Unfortunately, in practice these methods are rarely used because they often induce a large combinatory. This paper presents an optimization strategy to minimize the combinatory. The aim is to demonstrate the relevance of a competitive breadth-first exploration in off-line document analysis, in particular when the approach is interactive, ie the user is involved into the loop analysis. This paper demonstrates this optimized interactive analysis method on off-line handwritten 2D architectural floor plans.

Keywords—Sketch recognition; breadth-first exploration; interactive recognition; 2D architectural floor plans;

I. INTRODUCTION

We generally identified two major approaches to document analysis: syntactic and statistical approaches. The syntactic approaches [1] [2] lean on prior knowledge of the document structure to drive the analysis. They are often based on visual languages for describing this knowledge and generate the analyzer. The statistical approaches [3] [4] usually lack the ability to convey the hierarchical structure of the document and need a wide learning on a homogeneous and labeled base. While syntactic approaches can bring out the structural elements of the analysis document, statistical approaches provide a better ability to incorporate uncertainty. Our recent work [5] is focused on designing a new approach for interpreting documents, described as interactive. While most syntactic approaches are based on a depth-first exploration for interpreting structured document [2], we focused on the development of a breadth-first exploration allowing a confrontation, between several competitive hypotheses of interpretation for targeted context analysis. This strategy explicitly incorporate, if necessary, the user in the loop of analysis to raise ambiguities recognition [5], or to enrich the a priori knowledge of the system [6]. The user integration in the recognition loop avoids a fastidious a posteriori correction of errors recognition while allowing to have an auto-evolving system for interpreting sketches. Like all analysis methods based in breadth-first exploration, this approach induces a large combinatory.

In this paper, we focus on the optimization of the analysis strategy to reduce the engendered combinatory to obtain an analyzer that is really exploitable in the context of an interactive analysis; ie the user may be solicited by the analyzer during the interpretation phase to select the right hypothesis if the analyzer detects an ambiguity. This optimization is based on a dynamic limitation of the analysis tree branches depth. The analysis tree is optimally built at each stage of the document exploration. We validated this approach on the interactive analysis system for interpreting structured document (referred IMISketch) described in [5]. The experiments are made on off-line handwritten 2D architectural floor plans.

The remaining of the paper is organized as follows. In the section II, we introduce principles of our existing interactive analysis method for interpreting sketches "IMISketch". The optimization strategy of breadth-first exploration is described in section III. Experimental results are reported in section IV and finally, section V concludes the paper.

II. PRINCIPLES OF THE RECOGNITION SYSTEM IMISKETCH

In this section, we summarize the principles of our interactive method "IMISketch". This analyzer is based on the following characteristics:

- a priori structural knowledge of the document are expressed through a visual language based on production rules;
- a bidimensional descending breadth first analysis;
- the uncertainty is formalized by the attribution of scores to each hypothesis represented by the tree analysis branch;
- a spatial contextual focus of the exploration to limit the combinatory;
- if the ambiguities can not be resolved in the local context in an automatic manner, the user will be solicited by the analyzer to resolve the ambiguity.

These characteristics have been defined to ensure better interaction between the process of analysis and the user. This interactivity allows in particular to avoid a posteriori the verification phase, which can become fastidious on complex

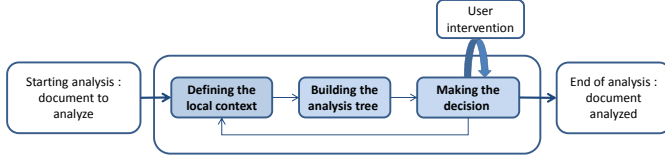


Figure 1. Analysis process

documents. Indeed, by appropriately soliciting the user on the critical phases of the analysis of the document, we avoid a cascade propagation of recognition error, which are very heavy to be a posteriori corrected. Figure 1 shows the complete process of analysis and the relationship between the three parts of the analyzer. The analyzer begins by defining a spatial contextual focus that aims to limit the combinatory exploration due to the breadth-first exploration of tree analysis. Once the context is well defined, the analyzer goes to build the analysis tree. In this stage, the analyzer develops all possible hypotheses of interpretation in the spatial context using a set of bidimensional rules that describe the structure of the document. These production rules are described by the context-driven constraint multiset grammars (CD-CMG) [7]. Each primitive can be interpreted in several ways. Each node or leaf is the application of a production rule deduced from the previous node. Every leaf or node of the tree has a score calculated from both its local score and the score obtained from the preceding nodes. Every score determines the adequacy degree to validate a production. The score calculated by each production is due to preconditions and constraints of the rule production (Equation 1). The use of the square root is a normalization using a geometric average. The production score can also be deduced from a classifier. A score is associated with each branch (hypothesis). Equation 2 determines the degree of adequacy (score) of a hypothesis. $|PS|$ is the number of production in the considered branch (referred as PS).

$$\rho_P = \sqrt{\mu_{preconditions} \cdot \mu_{constraints}} \quad (1)$$

$$\rho_{PS} = \left(\prod_{P_i \in PS} \rho_{P_i} \right)^{\frac{1}{|PS|}} \quad (2)$$

The analysis tree then contains a set of complete or incomplete objects. An object is called complete if and only if this object can be in the final result of the document interpretation. For example, in the case of the architectural plan analysis, the complete object can be walls, doors, windows... An incomplete objects are intermediate objects that can not exist in the final result of the analysis. In the case of the architectural plan analysis, we can find incomplete objects such as segment representing the beginning of an opening. Once tree analysis is explored, the analyzer starts the final phase: the decision making phase. The role of the decision process is to validate the right hypothesis among a set of competing hypotheses generated with a descending breadth first analysis. Sometimes the decision process is

not sure to make the right decision by validating the best hypothesis (because it has a score too low or it enters into confusion with the other hypotheses). In this case, it solicits the user. In practice, if the difference of scores between the top two branches is below a threshold of confidence and if these two branches are contradictory (at least one joint primitive is not consumed by the same rule production). When the correct root is validated, other roots are put on hold and the new roots are either the sons of this root if exists, or the waiting roots otherwise and the analyzer takes the first step (defining the local context step). The analysis is complete when no production rule is applicable. To assimilate this approach, we present in the following paragraph an example of interpretation of an architectural plan part.

A. Example of interactive breadth-first exploration

We present an example of image recognition of hand-written architectural plans consisting of walls, doors and windows. For this, we consider a set of production rules modeling architectural plans. These rules are illustrated in Figure 2. These rules express that at the end of a wall, we can find either a wall or an opening (door or window). An opening is located between two collinear walls. A sequence means an incomplete opening. We find in these rules, three complete objects: wall, door and window, and two incomplete objects: sequence and opening. In this example, the analysis is run on a set of primitives extracted from an architectural plan. After the segment extraction phase, the analysis defines the local context. The placement of this context in the document depends on the element consumed by the root interpretation. Figure 3 shows the movement of the local context between two phases of successive placement of the context. The construction phases of the tree deduced at this move are shown in Figure 4. The analyzer develops the tree as long as the following conditions (referred to as "IMISketch Conditions") are verified:

- 1) the newly consumed element is in the local context of research;
- 2) the number of consumed elements in each case (branch) is below a threshold;

Name	Created elements	Consumed elements
Ps	Starting wall	Longest line in the document
P1	Wall	line at the end of an other wall
P2	Sequence	line at the end of an interpreted wall or sequence
P3	Opening and wall	a sequence and two collinear walls or a sequence and wall and a line which are collinear
P4	Door	an opening
P5	Window	an opening

Figure 2. Example of production rules used for the architectural plans

Despite the use of local context in IMISkech, the problem of combinatory and therefore computing time, is an important problem that must be addressed to meet the criteria for acceptability and usability of this type of interactive analysis with a user. This problem is mainly due to the number of applied production rules in the local context. We note that the development of certain nodes in tree analysis is useless. We propose in the following section the different optimization strategies in the analysis process to solve the combinatory problem due to the development the tree analysis.

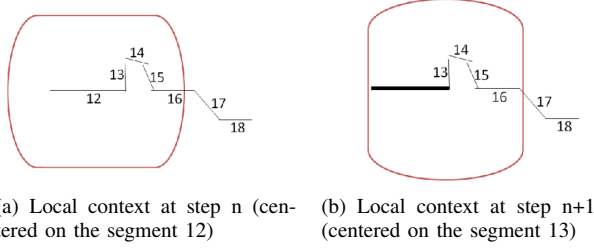
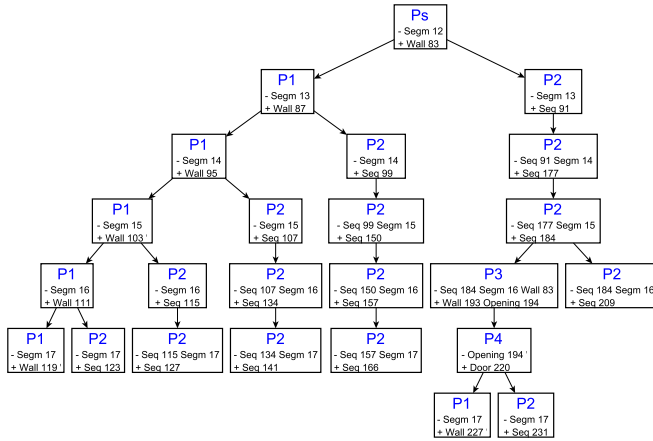
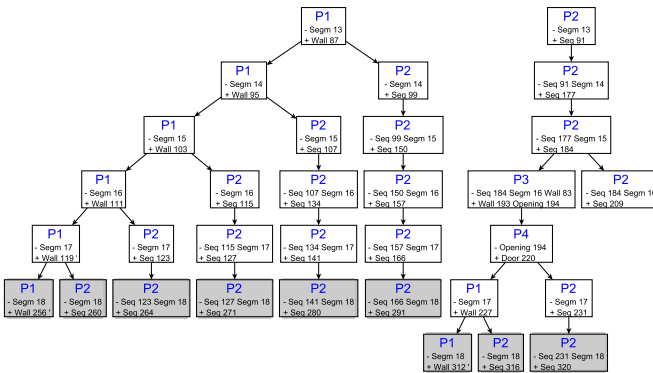


Figure 3. Document being analyzed and position of the local context (box) during the analysis



(a) Construction phase of tree analysis of stage n (segment 12) deduced from Figure 3(a)



(b) Construction phase of tree analysis of stage n+1 (segment 13) deduced from Figure 3(b). The grayed leaves indicate the new applied production rules in movement the local context.

Figure 4. The analysis trees associated with Figures 3(a) et 3(b). (-) and (+) designate respectively the consumed elements and the created elements.

III. STRATEGIES FOR REDUCING THE COMBINATORY

The construction of the tree based on a breadth-first exploration allows to have several competitive hypotheses. Unfortunately some production rules applied in a tree are invalidated by the system. In this section, we propose a new algorithm applied in the construction of tree analysis in order to develop the useful nodes for the right decision making. This algorithm is described below:

- If the number of analysis tree root is equal to 1:
 - 1) limit the development of direct son of the root.

If the root to develop is unique, we can say that only one interpretation is possible for the consumed element. The process of decision making is then sure to validate the right root (because it is unique). In this case, since we know the root to be validated, we consider unnecessary to build all tree analysis, and we can limit the tree development to direct son nodes. After validation of this root, these direct sons will be the new roots to build.
- If the number of analysis tree root is greater than 1:
 - 1) regroup the roots by consumed elements. We will not build all roots but only those that share the same elements. We want to find the right interpretation for consumed the element within the root.
 - 2) order these groups by their scores. Each group has a score derived from the roots within it. This score is the score of the best hypothesis (branch) located in the group.
 - 3) develop only roots belonging to the group having the best score.
 - 4) build analysis tree as long as the following conditions are verified:
 - the newly consumed element is in the local context of research (already used by IMISkech);
 - the number of consumed elements in each hypothesis (branch) is below a threshold (already used by IMISkech);
 - the number of complete elements belonging to a branch is less than a second threshold. We can fix a threshold to 3 for the architectural plans. The local context is not only limited to the distance between primitives but also to the number of complete elements in hypothesis. This optimization can generate a lack of information on hypotheses and therefore a possible ambiguity. But thanks to interactivity, this insufficiency does not influence on the final result as far as the user may be solicited to validate the right hypothesis.

All these optimizations allow to develop an interactive analysis of structured documents based on a in breadth-first

exploration of the tree analysis, by reducing the major problem of the breadth-first exploration: combinatory problem.

A. Comparison: IMISketch Vs optimized IMISketch

In this example, the aim is to show the improvements deduced from the new algorithm for building analysis trees. To facilitate this comparison, we present an example of an artificial tree analysis. The goal is to compare, for each step, the number of nodes developed by IMISketch and one induced by the new algorithm to build analysis tree. Figure 5(a) shows the result of the exploration according to the IMISketch method. The number of interpretation (nodes) is equal to 80 interpretations. The tree construction based on the new algorithm generates only 6 interpretations (Figure 5(b)). Figures 6(a) et 6(b) illustrate the new analysis trees by moving the local context. After two successive construction stages of analysis trees, we went from 111 interpretations (IMISketch) to 28 interpretations (optimized IMISketch), which has a gain of about 75% of computation time. These optimizations are generic and do not depend on the category of structured document to analyze. The following section presents the results quantified from a lot of 80 images of handwritten architectural plans.

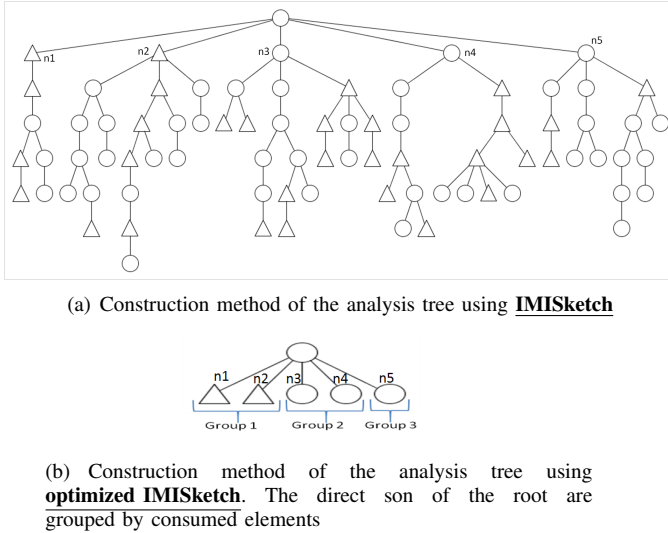
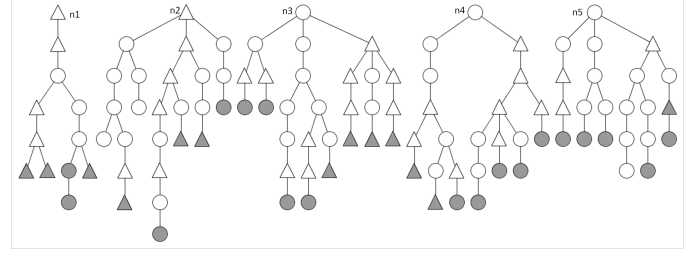


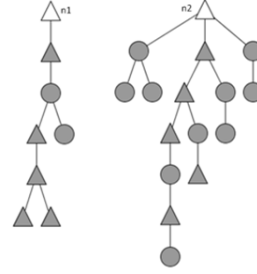
Figure 5. Analysis tree at step n. Nodes modeled by circles refer to complete object. Nodes modeled by triangles refer to incomplete objects.

IV. EXPERIMENTAL RESULTS

In this section we report results obtained from the different optimizations described in section III on the interactive method of document analysis. The aim of this optimization is to reduce the analysis time of the document to meet the criteria for acceptability and usability of the system. The experiments were carried on 80 architectural plans of varying complexity drawn by around ten different people. Each architectural plan is composed of a set of walls, doors, windows and sliding windows. Figure 7 shows the total



(a) Construction method of the analysis tree using **IMISketch**



(b) Construction method of the analysis tree using **optimized IMISketch**. The roots (n1, n2) belonging to the same group (Group 1) (same consumed elements)

Figure 6. Analysis tree at step n+1. Nodes modeled by circles refer to complete object. Nodes modeled by triangles refer to incomplete objects. The grayed leaves indicate the new applied production rules in movement the local context.

Number of architectural plans	80
Number of walls	3179
Number of doors	681
Number of windows	439
Number of sliding windows	509

Figure 7. Architectural plans symbols

number of symbols found within the 80 plans. Figure 8 shows some examples of analyzed architectural plans. We launched the analysis process on the 80 plans twice. The first time with the basic method (IMISketch) and the second time with optimizations presented in section III. For each architectural plan, we compare the final result of the interpretation and the obtained computing time. At the final

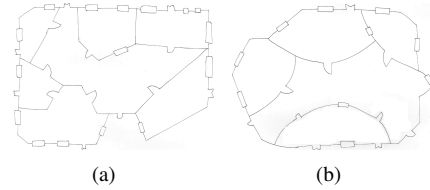


Figure 8. Examples of architectural floor plans

result of the interpretation, we find that the optimizations performed do not affect the system performance. Indeed, we observe a very light difference between IMISketch and optimized IMISketch in terms of recognition. We attain a recognition rate (without user solicitation) 95.25% with IMISketch and 96.17% with optimized IMISketch (see Figure 9). The user solicitation, during the analysis phase, can improve the recognition performance. Then, we attain 97.15% with IMISketch and 97.31% with optimized

IMISketch (see Figure 10). Figure 11 shows the solicitation percentage of users during the analysis loop versus the total number of interpreted roots and the useful solicitations percentage compared to the total solicitation, ie solicitations that allow improving recognition. Each element interpretation, the hypotheses proposed by the two methods (IMISketch and optimized IMISketch) are not the same (Figures 5 and 6). IMISketch produces more hypotheses than optimized IMISketch (see section III-A). The number of competing hypotheses is more important in IMISketch than optimized IMISketch, this might suggest that there have been more chance to have the right hypothesis with IMISketch, but the generated confusions are also potentially more numerous. In the end, one in the other, the results in terms of performance are very comparable. The found errors (about 3%) are due either to poor calibration of the local context, or the badly drawing of certain symbols. Figure 12 shows the difference of the computing time between IMISketch and optimized IMISketch for each architecture plan analysis. Depending on the complexity of the plans, the gain of computing time can reach 86% per image (image of index 62). We went from 16 hours and 22 minutes of accumulated computing time to 7 hours and one minute. This representing a gain of 57%. The experimental results are very encouraging. They suggest that it is possible to introduce descending breadth first analysis by controlling the generated combinatory. This supports the idea of conceiving an interactive systems for the document recognition. User solicitations, driven by analyzer, guarantee the obtaining of very high rate of reliability even when considering the complex documents treatment.

	IMISketch	optimized IMISketch
Number of unrecognized symbols	228	184
Recognition rate	95.25%	96.17%

Figure 9. Recognition rate on architectural plans **without** user solicitation

	IMISketch	optimized IMISketch
Number of unrecognized symbols	137	129
Recognition rate	97.15%	97.31%

Figure 10. Recognition rate on architectural plans **with** user solicitation

	IMISketch	optimized IMISketch
User solicitation	4.58%	3.77%
User effective solicitation	16.42%	12.06%

Figure 11. The percentage of user solicitation for 80 architectural plans

V. CONCLUSION

In this paper, we have presented an optimizing strategy for solving combinatory generated by an interactive analysis based on breadth-first exploration for the sketch recognition. This strategy is based on a dynamic construction of analysis tree by controlling the depth of each branch following a set of criteria specified in the current zone of interest of document. This optimization strategy has been validated and tested on an interactive analyzer for interpreting architectural plans. Note that this optimization approach is generic

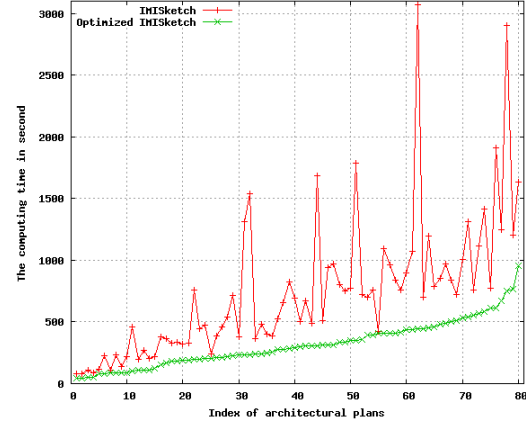


Figure 12. Gain of computing time per image

and therefore it could easily be applied to other types of structured documents, and other analyzers characterized by a breadth-first exploration.

Future work will focus on extending the experimental results on large image databases containing more complex architectural plans (integration of furniture, quotes, etc.)... We will also validate the criteria of acceptability and usability of the system by doing usage tests that will be conducted in collaboration with experts from the laboratory uses "Loustic" (<http://www.loustic.net/rennes>).

ACKNOWLEDGMENT

The authors would like to thank all the people who took part in the experiments. This work partially benefits from the financial support of the ANR Project Mobisketch.

REFERENCES

- [1] J. Fitzgerald, F. Geiselbrechtinger, and T. Kechadi, "Mathpad: A fuzzy logic-based recognition system for handwritten mathematics," in *ICDAR 2007*, vol. 2, 2007, pp. 694–698.
- [2] B. Coüasnon, "Dmos, a generic document recognition method: Application to table structure analysis in a general and in a specific way," *IJDAR 2006*, vol. 8, no. 2, pp. 111–122.
- [3] M. Lemaitre, E. Grosicki, E. Geoffrois, and F. Preteux, "Preliminary experiments in layout analysis of handwritten letters based on textural and spatial information and a 2d markovian approach," in *ICDAR 2007*, vol. 2, 2007, pp. 1023–1027.
- [4] F. Montreuil, E. Grosicki, L. Heutte, and S. Nicolas, "Unconstrained handwritten document layout extraction using 2d conditional random fields," *ICDAR 2009*, vol. 0, pp. 853–857.
- [5] A. Ghorbel, S. Macé, A. Lemaitre, and E. Anquetil, "Interactive competitive breadth-first exploration for sketch interpretation," *ICDAR 2011*.
- [6] A. Ghorbel, A. Almaksour, A. Lemaitre, and E. Anquetil, "Incremental learning for interactive for sketch recognition," *GREC 2011*.
- [7] S. Macé and E. Anquetil, "Eager interpretation of on-line hand-drawn structured documents: The dali methodology," *Pattern Recognition*, vol. 42, no. 12, pp. 3202–3214, 2009.